

20/402453.05

1 Docket No.: FX0-118.01

2
3 FIELDBUS UPGRADABLE APPARATUS AND METHOD

4
5 CLAIM OF PRIORITY

6 This application claims priority to provisional application
7 U.S.S.N. 60/136,253, entitled FIELDBUS UPGRADABLE DEVICE, and
8 filed on May 27, 1999, naming VLADIMIR KOSTADINOV as inventor,
9 the contents of which are herein incorporated by reference.

10
11 BACKGROUND OF THE INVENTION

12 (1) Field of the Invention

13 The present invention relates generally to communications
14 systems, and more particularly to utilizing communications
15 systems for uninterruptive software upgrading.

16 (2) Description of the Prior Art

17 In the manufacturing and process control industries, there
18 is a continuing effort to eliminate older, centralized plant
19 control strategies, in favor of standard protocols. This type of
20 control provides true device interoperability, enhanced field-
21 level control, and reduced installation costs.

22 The inherent modularity and intrinsic software reliance of
23 modern manufacturing and process control systems allows a
24 platform wherein distributed control may be best utilized.
25 Communications protocols such as the commonly known Fieldbus

1 technology allow for the interconnecting of measurement and
2 control equipment such as sensors, actuators, and controllers.
3 Examples of some Fieldbus technologies include Profibus™ and
4 Foundation™. Fieldbus is an all-digital, serial, two-way
5 communications system that serves as a Local Area Network (LAN)
6 for instruments in process control and manufacturing automation
7 applications. Fieldbus facilitates the distribution of the
8 control application across the network. Control through the
9 network is particularly advantageous when the devices or
10 processes to be controlled are physically remote from a central
11 control station.

12 Although the systems, sensors, and devices of such
13 manufacturing and process systems are modular, and there are
14 communications standards for interconnecting components of such
15 systems, there is currently no method or apparatus for allowing
16 device, system, or sensor software upgrades from remote locations
17 without interrupting the control system.

18 What is needed is a method and apparatus that facilitates
19 uninterrupted and remote upgrade of specified control system
20 components.

21 22 SUMMARY OF THE INVENTION

23 It is one aspect of the present invention to provide a
24 method and system for utilization with the well-known Fieldbus
25 communication protocol, that allows uninterrupted software

1 upgrading of remote, microprocessor controlled devices that exist
2 on the Fieldbus network.

3 In a preferred embodiment, the Fieldbus network comprises a
4 plurality of control devices, wherein control devices may be
5 input devices, output devices, or input/output devices. Each
6 control device may be configured for Fieldbus communications.
7 Each control device also comprises at least two memory areas
8 within which executable software and data may reside, wherein at
9 least one memory area may be active and at least one memory area
10 may be inactive. Memory areas may further comprise multiple
11 memory segments. Each control device additionally comprises a
12 selector device that specifies to the microprocessor those memory
13 areas that are active. In a preferred embodiment, the selector
14 device directs the control device microprocessor to the active
15 memory areas during the microprocessor operating system cycle.

16 It is another aspect of the present invention to allow a
17 Fieldbus communications system wherein a host computer configured
18 within the Fieldbus network may provide software upgrades and
19 other control to remote control devices through the network. In
20 an embodiment, software upgrades are performed without
21 interrupting the control device processor, by issuing a Fieldbus
22 compatible command that indicates to the control device
23 microprocessor that a new software version is available for the
24 designated control device. The respective microprocessor routes
25 the software upgrade to a presently inactive memory area within

1 the designated control device that shall be designated the new
2 memory area. Such transfer occurs during otherwise unscheduled
3 communications periods to avoid interruption to the network or
4 the control device. Similarly, the remotely located control
5 device microprocessor performs the transfer without interrupting
6 the presently executing application or data functions in the
7 control device. The microprocessor also verifies the new
8 software. The new memory area may be activated during a
9 microprocessor idle period by directing the microprocessor to the
10 new software, thereby providing a seamless transition to the new
11 memory area and hence the new software.

12 It is another aspect of the invention to provide a mechanism
13 whereby the host may return the microprocessor to a previously
14 active memory area within a designated control device.

15 It is yet another aspect of the invention to utilize new
16 memory areas for increasing the control device functionality.
17 The number of active memory areas may increase as functionality
18 is increased.

19 Other objects and advantages of the present invention will
20 become more obvious hereinafter in the specification and
21 drawings.

1 BRIEF DESCRIPTION OF THE DRAWINGS

2 A more complete understanding of the invention and many of
3 the attendant advantages thereto will be readily appreciated as
4 the same becomes better understood by reference to the following
5 detailed description when considered in conjunction with the
6 accompanying drawings, wherein like reference numerals refer to
7 like parts and wherein:

8 FIG. 1 is a representative block diagram of a Fieldbus
9 communications network segment displaying a host and a single
10 control device;

11 FIG. 2 presents a control device initialization methodology;

12 FIG. 3 diagrammatically represents the logic for altering
13 control device memory; and,
14 FIG. 4 presents the microprocessor operating system and

15 unscheduled communications timelines.
16

17 DESCRIPTION OF THE PREFERRED EMBODIMENT

18 To provide an overall understanding of the invention,
19 certain illustrative embodiments will now be described; however,
20 it will be understood by one of ordinary skill in the art that
21 the systems described herein can be adapted and modified to
22 provide systems for other suitable applications and that other
23 additions and modifications can be made to the invention without
24 departing from the scope hereof.

1 Referring now to FIG. 1, there is shown a block diagram of a
2 system 10 that incorporates an embodiment of the invention. The
3 FIG. 1 system 10 comprises a host 12 that is configured for
4 communications using the well-known Fieldbus communications
5 protocol. The host 12 can be implemented using a digital
6 computer system that may be any microprocessor-based system
7 including a computer workstation, such as a PC workstation, SUN
8 workstation, handheld or laptop computer, that comprises a
9 program for organizing and controlling the digital computer
10 system to operate according to the invention. Additionally and
11 optionally, the microprocessor-based system can be equipped for
12 processing multimedia data, and can be, for example, a
13 conventional PC computer system with a sound and video card. The
14 computer system can operate as a stand-alone system when not
15 operating as part of a networked computer system. The host may
16 therefore be any microprocessor-based device that is designated
17 to perform the functionality herein to uninterruptively upgrade
18 software on a specified control device, such specified control
19 device including, for example, a field device such as a
20 controller, transmitter, or actuator.

21 The host 12 may provide and receive remote command and
22 control information to and from a plurality of control devices
23 residing on the network. FIG. 1 displays a representative
24 control device 14. The host 12 and control device 14 may not be
25 co-located on the same Fieldbus segment, and communications

1 between the host 12 and control device 14 may be coupled through
2 various Fieldbus and other network segments, that shall be
3 referred to herein collectively as the Fieldbus communications
4 network.

5 Control devices may be input devices, output devices, or
6 input/output devices as commonly known in the art. As indicated
7 by FIG. 1, the control device comprises a Fieldbus communications
8 interface 16 to receive and transmit commands and data across the
9 Fieldbus network, at least one microprocessor 18 to execute
10 control device resident applications and communicate with the
11 host and other control devices, at least two memory areas 20, 22
12 for storing executable programs and/or data that may be accessed
13 by the microprocessor 18, and a selector device 24 interfaces the
14 microprocessor 18 to executable instructions or data in the
15 memory areas 20, 22.

16 In a preferred embodiment, the memory areas 20, 22 may be
17 flash memory, although other memory devices may be utilized
18 without departing from the invention. Each memory area 20, 22
19 may additionally be continuous, partitioned, or segmented. The
20 FIG. 1 memory areas 20, 22 therefore merely represent logically
21 separate memory areas, and each memory area 20, 22 may be
22 comprised of memory across multiple segments, etc.

23 For the purposes of this invention, all references to the
24 microprocessor 18 shall be understood to refer to the embedded
25 software or operating system that forms a microprocessor

1 operating system, wherein such an operating system is commonly
2 known for scheduling and executing applications, allocating
3 resources, etc. In a preferred embodiment, the microprocessor 18
4 repeatedly performs a scheduled series of tasks during a fixed
5 time period. The tasks are identified to the microprocessor 18
6 through entry points that specify a memory area, wherein the
7 memory area comprises executable instructions or data to perform
8 and/or complete the task. In an embodiment, the memory area may
9 comprise executable instructions or data relating to a control
10 device application.

11 For purposes of this invention, each such repetitive fixed
12 time period wherein the microprocessor performs the scheduled
13 tasks shall be defined as a macrocycle. Additionally, any time
14 within the macrocycle during which the microprocessor is not
15 executing an application, performing application-dependent
16 input/output, or performing application related communications,
17 shall be defined as an idle period.

18 As FIG. 1 indicates, the microprocessor 18 may access the
19 memory areas 20, 22 in two manners for two distinct purposes.
20 The microprocessor connection to the memory areas 20, 22 through
21 the selector device 24 represents the selector device
22 functionality to supply the microprocessor 18 with entry points
23 for application executable instructions or data, wherein such
24 application instructions or data resides in the memory areas 20,
25 22. For the purpose of this invention, such activity shall be

1 defined as "executable" activity, wherein the microprocessor 18
2 executes the instructions or data residing in the memory areas
3 20, 22.

4 Alternately, the microprocessor 18 may directly access the
5 memory areas 20, 22 (i.e., without selector device 24
6 intervention) to perform functions unrelated to application or
7 data execution. Examples of such functions include data
8 integrity checks, data loading or unloading, etc. For purposes
9 of this invention, such activity shall be defined as "processing"
10 activity, wherein the microprocessor 18 processes the memory area
11 contents without executing the instructions or data therein.

12 The selector device 24 may designate an active memory
13 area(s) from an inactive memory area(s) for the microprocessor
14 18. For the purposes of this invention, active memory areas
15 shall be defined as the memory areas that the microprocessor 18
16 is directed to, by the selector device 24, to obtain executable
17 instructions or data. In a preferred embodiment, the selector
18 device 24 activates a memory area merely by providing the
19 microprocessor 18 with the entry points to the memory area.

20 For purposes of this invention, all memory areas other than
21 the active memory areas shall be known as inactive memory areas.

22 The microprocessor 18 does not execute instructions or data from
23 any inactive memory area, however the microprocessor may process
24 the inactive memory area contents for data integrity, perform
25 data downloading, etc.

1 For purposes of this invention, a new memory area shall be
2 defined as an inactive memory area to which upgradeable
3 executable instructions (e.g., an application) or data shall be
4 directed, wherein such upgradeable instructions or data shall
5 also be referred to collectively as new data. New data may work
6 independently, to the exclusion of, or together with, existing
7 data in active memory areas.

8 In a preferred embodiment, the selector device 24 is
9 incorporated as a software module that interacts with the
10 microprocessor 18 and may be implemented through software using
11 higher-level languages such as C++ or Java, or optionally
12 microcode or machine level instructions; however, those with
13 ordinary skill in the art shall recognize that the selector
14 device 24 may be implemented in hardware without departing from
15 the scope of the invention herein.

16 Referring now to FIG. 2, there is shown a representative
17 process 40 by which a control device may be initialized. In an
18 embodiment presented in FIG. 2, upon initialization, the control
19 device cycles through all control device memory areas and sets to
20 active all memory areas that are verified through the
21 verification process. In such an embodiment, the control device
22 microprocessor selects, in a logical order, a memory area 42.
23 The microprocessor may then verify 44 the memory area contents
24 using such well-known techniques as CRC computations, although
25 the invention is not limited by such verification technique. If

1 the verification fails, the memory area may be marked as inactive
2 46, and another memory area is selected 42. In a preferred
3 embodiment, such inactive designation 46 may be a passive
4 activity since all memory areas other than the active memory area
5 may be by default, inactive; however, alternate embodiments may
6 otherwise designate inactive memory areas using alternate
7 techniques without departing from the invention herein.

8 Alternately, when a memory area is properly verified, a data
9 area may be assigned 48 within the memory area, and the memory
10 area may be designated active 50. The next memory area in the
11 logical order may then be selected 42, until all such memory
12 areas are designated active or inactive.

13 Although FIG. 2 presents an initialization process for one
14 embodiment wherein multiple memory areas may be active, alternate
15 embodiments may utilize different initialization processes. In
16 applications wherein only one memory area may be allowed to be
17 active, such memory area may be "selected" 42 by the respective
18 selector device or microprocessor that may retain or preserve the
19 active and inactive memory area status data from a previous
20 session. In yet another embodiment, the host may store the
21 location of the active memory areas for each control device, and
22 transmit, using the Fieldbus protocol, the active memory
23 address(es) to the respective control device upon initialization.

24 Referring now to FIG. 3, there is shown a process 60 wherein
25 new executable instructions or data may be provided to a

1 designated control device. In the FIG. 3 embodiment, updating
2 the control device comprises downloading new data. The host may
3 request an upgrade to the control device, and the request may be
4 communicated between the host and control device using Fieldbus
5 protocols. As FIG. 1 indicates, communications between the host
6 and the control device require communication through the Fieldbus
7 interface, that thereafter interacts with the microprocessor that
8 processes the commands. Such communications may occur during
9 otherwise unscheduled communications periods between the host and
10 control device, wherein unscheduled communications periods may be
11 defined as those time periods during which the control device
12 microprocessor may not be previously scheduled to exchange
13 input/output data relating to applications executing on the
14 control device. Such previously scheduled input/output periods
15 may comprise communications between the control device and the
16 host, or between the control device and another control device.

17 The host may identify the control device through a user-
18 interface or other interactive mechanism that allows a host
19 operator to designate a specific control device. In a preferred
20 embodiment, the host comprises a user interface that similarly
21 indicates respective active and inactive memory areas for
22 specified control devices. The host may store information
23 regarding the present active memory areas for a specified control
24 device and display such information to the host user, or
25 alternately, the host may poll the selector device of the

1 specified control device, using the Fieldbus protocol, to
2 ascertain the present active memory areas. In yet another
3 embodiment, the selector device may provide the host with the
4 active memory area designation upon completion of the control
5 device initialization processing as indicated by FIG. 2.

6 In a preferred embodiment, the host user interface may allow
7 a host operator to specify an inactive memory area as a target
8 for the new data. Such user interface may also allow the host
9 operator to specify the new data. Referring back to FIG. 3, the
10 host may then issue, using Fieldbus communications protocols, a
11 data download request 62 that may be received by the specified
12 control device microprocessor to cause the new memory area
13 identifier and new data to be transmitted to the control device.

14 In a preferred embodiment, the host also transfers the new
15 memory area entry points, wherein the entry points shall direct
16 the microprocessor to the executable instructions or data in the
17 new memory area. Such host requests and transfers are also
18 performed during previously defined unscheduled communications
19 intervals.

20 Upon receipt 64 of the download request and new memory area
21 designation, new data, and new memory area entry points, the
22 microprocessor may direct the new data to the new memory for
23 storage. The microprocessor may also direct the new memory entry
24 points to the selector device for storage. The host may then
25 issue a verification command that may cause the control device

1 microprocessor to verify 66 the new data in the new memory area,
2 wherein such verification may be, for example, a CRC computation,
3 but the invention herein is not limited to such verification
4 method. In an embodiment, the microprocessor stores the result
5 of such memory verification in the selector device. In a
6 preferred embodiment, the microprocessor functions of receiving
7 64, redirecting 64, and verifying 66 the new data, may be
8 performed in parallel processes with the presently executing
9 applications in the active memory areas. Such functions may
10 therefore be performed at any time in the microprocessor
11 timeline, other than the during scheduled input/output network
12 communications intervals.

13 If a proper verification of the new memory area is not
14 achieved 68, the host may issue another download request, again
15 designating a new memory area and the new data. The same
16 inactive memory area may be utilized, or the host may request a
17 different memory area if more than one inactive memory areas
18 exist.

19 Alternately, upon proper verification of a memory area, the
20 control device microprocessor may inform the host of the proper
21 verification, wherein the host may request that the new memory be
22 designated an active memory 70.

23 The control device microprocessor, upon receiving a request
24 from the host to change the status of an inactive (i.e., new)
25 memory area to active, may interrogate the verification status of

1 the new memory 72. Alternately, the microprocessor may
2 interrogate whether entry points are received for the new memory
3 area. If the new memory area is not verified, or entry points do
4 not exist, the microprocessor may reject 74 the host request to
5 change the active memory designation. Alternately, if the new
6 memory is verified and entry points are stored, the host's
7 request may be granted, and the microprocessor may inform the
8 selector device that the new memory area may be activated 76.

9 Depending upon, for example, whether the new data comprises
10 cooperative as compared to replacement data, the host may issue a
11 request to inactivate a presently active memory area 78. In some
12 embodiments, activating and inactivating memory areas may require
13 proper coordination and timing to achieve the desired
14 microprocessor direction via the selector device-provided entry
15 points.

16 In one embodiment wherein the system is initialized through
17 the cyclic process depicted in FIG. 2, memory inactivation may
18 cause the memory area to purposefully fail subsequent
19 verification checks. In such a system, re-activating the content
20 of that memory segment may require a new download of the older
21 executable instructions or data. Alternately, the inactivation
22 process may be reversible with another command from the host,
23 thereby preventing the necessity for another data download.

24 In embodiments wherein only one memory area can be active,
25 the inactivation process may be simplified and may be implemented

1 using default logic embedded in software or hardware. All such
2 activation and inactivation methods and processes may be
3 implemented in various manners without departing from the scope
4 of the present invention.

5 The method of indicating active or inactive memory areas
6 relates also to the ability to return to previous versions of
7 executable instructions or data. For example, if a system
8 comprises N memory areas, wherein only one memory area may be
9 active, the system may be implemented such that as many as (N-1)
10 versions of executable instructions or data may be stored in
11 inactive memory areas. Such configuration may also allow rapid
12 transition between these versions of executable instructions or
13 data by issuing a host request with the new memory area
14 designation.

15 Alternately, when multiple memory areas are allowed to be
16 active, the present invention provides a structure wherein
17 functionality may be easily added or eliminated. Modular
18 upgrades may also be readily achieved in such an embodiment,
19 wherein one memory area with a specific functionality may be
20 upgraded to the exclusion of other active memory areas.

21 Referring now to FIG. 4, there is shown two timelines 80
22 indicating a representative microprocessor operating system
23 timeline, previously defined as a macrocycle, and a corresponding
24 unscheduled communications timeline. As mentioned previously,
25 the macrocycle 82 may be viewed as a repeatable interval, the

1 duration of which may be designed to allow execution of all
2 control device applications and input/output functions, with
3 consideration for the overall communications bandwidth during the
4 input/output functions. For example, during each representative
5 macrocycle of FIG. 4, the microprocessor operating system
6 executes a first application 84, wherein such first application
7 is followed immediately by a first input/output interval 86.
8 During the first input/output interval 86, the control device may
9 transfer data relating to the first application, using the
10 Fieldbus protocol. Such communications intervals 86 are
11 therefore known as scheduled communications intervals, and may be
12 between the control device and the host, or between the control
13 device and another control device. Upon completion of the first
14 input/output interval 86, the microprocessor executes a second
15 application 88, wherein such second application 88 is followed in
16 time by a second input/output interval 90, such second
17 input/output interval 90 also being a scheduled communications
18 interval. Although the representative embodiment of FIG. 4
19 indicates only two applications, the invention herein is not
20 limited by the number of applications executed on a control
21 device during a given macrocycle.

22 Referring again to FIG. 4, the selector device may activate
23 or inactivate memory areas, or change the entry points, at any
24 time during the macrocycle that the microprocessor operating
25 system is not executing applications or utilizing application

1 dependent data. Such periods in the macrocycle are indicative of
2 the previously defined idle periods 92. During these idle
3 periods 92, the selector device may incorporate the entry points
4 to the new memory area such that during the microprocessor's next
5 scheduled application execution, the new entry points may direct
6 the microprocessor to the new data. By altering the
7 microprocessor entry points during intervals wherein the
8 microprocessor is not performing application or application-
9 dependent processing, the transition to the new data may occur
10 without interrupting the microprocessor functionality or
11 processing timeline.

12 As mentioned previously, activating a memory area may
13 require inactivating another memory area, and such
14 activation/inactivation may require coordination. Inactivation,
15 just as activation, must be performed during the idle periods 92,
16 when the microprocessor is neither scheduled for application
17 execution or application input/output. Such
18 activation/inactivation may require several idle periods to
19 achieve complete conformance. In an embodiment, the host may
20 extend an idle period (i.e., prevent the next macrocycle from
21 commencing) to effectuate a memory area activation or
22 inactivation.

23 Alternately, data transfers from the host computer to the
24 new memory area, or any commands or requests from the host, may
25 be scheduled during the unscheduled communications periods 94.

1 As defined previously, unscheduled communications periods 94
2 comprise any macrocycle time interval during which the
3 microprocessor is not performing input/output with the host or
4 another control device. Unscheduled communications therefore
5 comprise all time within the processing interval that is not a
6 scheduled communications interval.

7 As with any communications system, data rates must be
8 considered when scheduling events. Depending upon the data rates
9 and the data amount transferred to a new memory area, more than
10 one macrocycle may be required to download the data; therefore,
11 the download of executable instructions or data to a new memory
12 area may occur during one macrocycle, while the verification of
13 such new memory may not occur until several macrocycles after
14 such download begins. Further, altering the microprocessor
15 operating system entry points may require a period greater than a
16 single idle period 92, thereby delaying the activation of a new
17 memory area for several macrocycles.

18 In one embodiment where several control devices may be
19 upgraded, the upgrades may be coordinated to occur at the same
20 time. In such an embodiment, new data may be downloaded to all
21 control devices, and respective microprocessors may be redirected
22 to respective new data memory areas to achieve a synchronized or
23 otherwise controlled upgrade.

24 In one embodiment, the microprocessor redirection for one or
25 more control devices may be scheduled to a certain time or event,

1 without departing from the invention. In such an embodiment, one
2 or more control devices may comprise new data in an inactive
3 memory area. The host may then monitor at least one parameter,
4 wherein the parameter(s) may relate to control devices, and upon
5 the particular parameter(s) attaining a predetermined value, the
6 host may issue a request to redirect the microprocessor on one or
7 more control devices.

8 The advantage of the present invention over the prior art is
9 that control devices within a Fieldbus network may be remotely
10 updated with new executable instructions or data without
11 disturbing the operation of the control device.

12 What has thus been described is a method and apparatus to
13 modify control devices residing on a Fieldbus communications
14 network, without interrupting the operation of the control
15 devices. The control device updating may further be controlled
16 and monitored by a remotely located host that also communicates
17 on the Fieldbus network. The control device may comprise at
18 least two distinct memory areas, wherein at least one memory area
19 must be active, and at least one memory area must be inactive.
20 Active memory areas provide the control device microprocessor
21 operating system with executable instructions or data. The host
22 downloads new executable instructions or data to inactive memory
23 areas, with associated data entry points, during unscheduled
24 communications periods wherein data input/output is not being
25 performed between the control device and the host or another

1 control device. Upon a full data transfer and proper
2 verification of the new data, the host may issue an activation
3 command that causes a selector device to activate the previously
4 inactive memory area by directing the microprocessor to the entry
5 points of the newly downloaded executable instructions or data.
6 The memory activation must occur while the microprocessor is not
7 performing application execution, application input/output, or
8 application communications. By timing the memory activation in
9 this manner, the microprocessor may be redirected to the newly
10 downloaded executable instructions or data without microprocessor
11 interruption.

12 Although the present invention has been described relative
13 to a specific embodiment thereof, it is not so limited.
14 Obviously many modifications and variations of the present
15 invention may become apparent in light of the above teachings.
16 For example, the selector device functionality may be performed
17 in hardware or software. The selector device may be incorporated
18 within the microprocessor or independent of the microprocessor
19 operating system. The Fieldbus network may contain any number of
20 control devices. Each control device may have a different
21 macrocycle length during which a varying number of applications
22 may be executed. Depending on the network size, there may be
23 more than one host. The host and the control devices may reside
24 on different Fieldbus segments, wherein such segments may be
25 connected through otherwise compatible network software or

1 hardware. The interactions and scheduling between the
2 microprocessor and the selector device may be embedded in either
3 system or otherwise shared between the systems. Wherein multiple
4 memory areas may be active and multiple memory areas are
5 inactive, multiple memory areas may be updated and all
6 corresponding entry points changed within the same selector
7 device modification.

8 Many additional changes in the details, materials, steps and
9 arrangement of parts, herein described and illustrated to explain
10 the nature of the invention, may be made by those skilled in the
11 art within the principle and scope of the invention.

12 Accordingly, it will be understood that the invention is not to
13 be limited to the embodiments disclosed herein, may be practiced
14 otherwise than specifically described, and is to be understood
15 from the following claims, that are to be interpreted as broadly
16 as allowed under the law.